

**REMARKS**

Claims 1-48 are pending in the instant application. The Examiner has allowed claims 32 and 48. Claims 1 and 17 stand rejected under 35 U.S.C. 102(b) as allegedly being anticipated by Juergen Boldt, "Complete MOF 1.3 specification", document: formal/00-04-03 (MOF 1.3) (hereinafter Boldt). Claims 13-15, 29-31, 33, and 45-47 stand rejected under 35 USC 102(b) as allegedly being anticipated by U.S. Patent No. 5,854,929 to Van Praet et al., (hereinafter Van Praet). Claims 2-7, 10-12, 18-23, 26-28, 34-39, and 42-44 stand rejected under 35 U.S.C. 103(a) as being allegedly unpatentable over Boldt as applied to claims 1, 17, and 33, in view of Van Praet. Finally, claims 8, 9, 16, 24-25, and 40-41 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Claims 1-48 have been canceled and New claims 49-72 have been added rendering the rejections of claims 1-48 moot. The Applicant submits that the instant application is in condition for allowance for at least the reasons presented herein.

**Claims 49-53, 57-61, and 65-69**

New claims 49, 57, and 65 recite, respectively, a method, storage medium and signal for compiling source code written in an object-oriented source language into executable machine-language code for a target computer, comprising: "defining in said source code, using said source language, software-visible hardware objects of said target computer as pre-existing objects of classes defined in said source language, wherein at least one of member methods of said classes contain statements in said source language, said statements containing expressions evaluating to values of operation codes and associated operand specifiers in said machine-language; and

compiling results of said defining into said machine-language code by expanding, inline, every invocation of said member methods of said classes to said values of

TAJ-0001

operation codes and associated operand specifiers to which their expressions evaluate, said expanding inline of invocations continuing recursively until all invocations of said member methods have been replaced by operation codes and associated operand specifiers, said compiling accomplished without requiring application of an intermediate language." The software-visible physical objects and methods for compiling have been clarified and presented in a non-narrowing manner to clarify that which the Applicant regards as the invention. The limitations recited in New claims 49, 57, and 65 differentiate between the physical objects recited in Boldt and the use of an instruction set specific to a target architecture as provided in Van Praet. The Applicant submits that New claims 49, 57, and 65 are in condition for allowance. Claims 50-53, 58-61, and 66-69 depend from what are now allowable claims 49, 57, and 65. For at least these reasons, the Applicant submits that claims 50-53, 58-61, and 66-69 are also in condition for allowance.

Claims 54, 62, and 70

New claims 54, 62, and 70 recite, respectively, a method, storage medium, and signal for compiling literals in a source code, written in an object-oriented source language, into at least one of data objects and executable machine-language code for a target computer, comprising: "defining in said source code, using said source language, classes having initializer member methods that take as arguments an internal representation of said literals of a compiler, said initializer member methods previously compiled from said source code; upon parsing one of said literals, said compiler calling one of said initializer member methods, said initializer member method taking said internal representation of said parsed literal as an argument resulting in an initialized object; and said compiler performing at least one of: incorporating said initialized object into an output of said compiler; and using said initialized object in subsequent compilation steps." The compilation of literals recited in the Applicant's New claims 54, 62, and 70 are further defined and presented in a non-narrowing manner to clarify that which the Applicant regards as the invention. The limitations recited in New claims 54,

62, and 70 specify the defining of "classes having initializer member methods that take as arguments an internal representation of said literals of a compiler, said initializer member methods previously compiled from said source code...said compiler calling one of said initializer member methods, said initializer member method taking said internal representation of said parsed literal as an argument resulting in an initialized object". Van Praet discloses invoking statements in the process of compilation; however, Van Praet does not teach or suggest defining classes having initializer member methods (previously compiled from source code) that take as arguments an internal representation of the literals of a compiler and calling one of the initializer member methods, that takes the internal representation of the parsed literal as an argument resulting in an initialized object. Accordingly, the Applicant submits that New claims 54, 62, and 70 are in condition for allowance.

Claims 55, 56, 63, 64, 71, and 72

New claims 55, 63, and 71 recite, respectively, a method, storage medium, and signal for compiling two object-oriented classes from a single class literal in a source code, comprising: "in said source code, labeling at least one member method defined in said single class literal as belonging to a constant version of a class, said constant version of said class differentiated from a variable version of said class; and said compiler generating two classes from said single class literal, said constant version of said class being a base class of said variable version of said class." Likewise, New claims 56, 64, and 72 recite, respectively, a method, storage medium, and signal for compiling two object-oriented class interfaces from a single class interface literal in a source code, comprising: "in said source code, labeling at least one member method defined in said class interface literal as belonging to a constant version of said class interface, said constant version of said class interface differentiated from a variable version of said class interface; and said compiler generating two class interfaces from said single class interface literal, said constant version of said class interface being a base class of said variable version of said class interface." Neither reference, Boldt nor Van Praet, teach or

TAJ-0001

suggest compiling two object-oriented classes from a single class literal in a source code or compiling two object-oriented class interfaces from a single class interface literal in a source code. Accordingly, the Applicant submits that claims 55, 56, 63, 64, 71, and 72 are in condition for allowance.

In view of the foregoing, it is respectfully submitted that the instant application is in condition for allowance. Accordingly, it is respectfully requested that this application be allowed and a Notice of Allowance issued. If the Examiner believes that a telephone conference with Applicant's attorneys would be advantageous to the disposition of this case, the Examiner is cordially requested to telephone the undersigned.

No New matter has been entered and no additional fees are believed to be required. In the event the Commissioner of Patents and Trademarks deems additional fees to be due in connection with this application, Applicant's attorney hereby authorizes that such fee be charged to Deposit Account No. 06-1130 maintained by Applicant's attorneys.

Respectfully submitted,

CANTOR COLBURN LLP

By Marisa J. Dubuc  
Marisa J. Dubuc  
Registration No. 46,673  
Customer No. 23413

Date: December 22, 2004  
Address: 55 Griffin Road South, Bloomfield, CT 06002  
Telephone: (860) 286-2929

TAJ-0001